

Q1. Are all five functions defined in section 14 to be included in the SDKs for each class, even though, as an example, two of the functions will never be used for class X submissions?

A1. Please see table below.

Class	What to submit	API to use	What to evaluate
X	quality algorithm	<code>compute_quality_from_image_data()</code>	Class X quality algorithms are evaluated using class Z matchers.
Z	matching algorithm and proprietary template generator	<code>convert_image_to_proprietary_template()</code> <code>match_proprietary_templates()</code>	Class Z comparison algorithms are used to evaluate class X or Z quality algorithms.
Z	quality algorithm, matching algorithm and proprietary template generator	If quality computation is part of template generation: <code>convert_image_to_proprietary_template()</code> <code>match_proprietary_templates()</code> if standalone quality algorithm <code>compute_quality_from_image_data()</code> <code>convert_image_to_proprietary_template()</code> <code>match_proprietary_templates()</code>	Class Z comparison algorithms are used to evaluate class X or Z quality algorithms.
Y	quality algorithm + mated matching algorithm and proprietary template generator	If quality computation is part of template generation: <code>convert_image_to_proprietary_template()</code> <code>match_proprietary_templates()</code> if standalone quality algorithm <code>compute_quality_from_image_data()</code> <code>convert_image_to_proprietary_template()</code> <code>match_proprietary_templates()</code>	A class Y quality algorithm is ONLY evaluated against its mated comparison algorithm.

If you want your quality algorithm be evaluated against your matching algorithm only, you are a class Y participant. Class Y matcher is not used to evaluate class X or class Z quality scores.

If you want your quality algorithm be evaluated against all possible (class Z) matchers, and your matcher be used for analysis and evaluation of other class X or Z quality algorithms you are a class Z participants.

Q2. What APIs should I use?

A2. Please see table above.

Class Y and Z use these APIs:

- if quality computation is part of template generation:
`convert_image_to_proprietary_template()`, and
`match_proprietary_templates()`
- if quality computation is not part of template generation
`compute_quality_from_image_data()`,
`convert_image_to_proprietary_template()`, and
`match_proprietary_templates()`

Class X participants use `compute_quality_from_image_data()`.

Q3.1. Why allow quality to be reported either from `convert_image_to_proprietary_template()` or `compute_quality_from_image_data()` for class Y?

A3.1 This is to support operationally relevant cases where quality is computed as part of template generation.

Q3.2. If both report quality, will both be assessed? Will assessments of speed rely on one rather than the other?

A3.2. We will report quality computation time for standalone quality (class X) and quality as part of template generation (class Y or class Z) separately.

Q4. Why make reporting quality optional for class Z?

A4. Reporting quality is optional for class Z to allow wider participation. Some organization may only have matching algorithm and some (specially academic institutions) may only have quality algorithm and no matching capabilities. By allowing submitting only quality or only matcher or both, we are facilitating and encouraging a wider participation.

Q5. Would it be acceptable to submit only in class Z and report no quality measures?

A5. Yes

Q6. Would it make sense to submit in all three classes?

A6. No

Q7. Will the results of the testing of a class Y submission be a subset of those for a class Z submission with quality output?

A7. No